1)PythonCI

Implementated complete CI process on python-django framework.

Details of tools used-

a) Bamboo as CI-server,

b)coverage for code coverage,

c)cobertura\_clover\_transform.converter for conversion of reports to clover.xml to be rendered in bamboo

d)pylint for statis code analysis

e)epydoc for api documentation

f)executed sonar-runner from shell-script

g)"Script" task of bamboo to automate the CI process.(shell-scripting)

h)GIT for SCM.

i)Stash for repository management

2)Php-CI

-Implemented PhpCI on 59.54 server + on local VM from scratch(including environment creation and plugins installations in Jenkins). Details of tools used-

a)Jenkins as CI server

b)ANT as build-tool for automating the CI process

c)GIT for SCM.

d)Stash for repository management

e)For rendering reports of static code analysis in jenkins used "checkstyle", "pmd", "duplicate code analysis" results plugins.

f)"Phpunit-3.x" for rendering test-results

g)"Clover PHP coverage report" for rendering code-coverage results

h)SonarQube integration for static code analysis results

i)JSlint and csslint for code-analysis of js and css files respectively.

j)Phpdox for rendering reports of API docementation.

3)JSlint and CSSLint

-exploration and implementation in PhpCI

-Used Jenkins as CI server.

-Used "checkstyle" plugin for displaying csslint reports.

-Used "JSHint warnings" plugin for displaying jslint reports.

-Also displayed jslint and csslint reports using "Violations" plugin.

4)Php-CD

-Jenkins as CI server

-Performed DEPLOYMENT using scp,ssh from VM to 59.62 server.

-Performed deployment of package generated in ci process using jenkins on remote server from VM triggered from jenkins using ant shell-script.

-Implemented Delivery pipelins as well as build-pipeline(CI+CD as upstream and downstream jobs)

5)Xdebug

-Implemented profiling of PHP scripts using Xdebug.

-Rendered these profiling reports in a dashboard using the tool "webgrind"(used linux)

6)GULP

1. Understanding the basics of Gulp build tool.

2. Exploring difference from Grunt and how/why better.

3. Understanding Node.js, NPM and the node modules.

4. Learning and documenting all CI tasks for Javascript/PHP codebase.

5. Exploring important CI Gulp plugins required and how/where to use them.

6. Exploration of the gulp build script and implementing how to add tasks and pipes for build flow.

7. Exploration and implementation of Sonar plugin and PHP Static Code Analysis tools.

8. Implementation of CI tasks like Minify, Uglify, PHPUnit Test cases, PHP Document creation

9. Exploring plugins for CI process for php (unit-testing, code-coverage, static code analysis).

10. CI implementation for php.

11. Implemented Browser-Sync.(special feature)

12. Divided a single Gulp file into two parts. One is dedicated for CI implementation and the other for development.

13. Implemented phpmd.

14. Implemented Copy Paste detector.

15. Included htmllint and also gulp-concat.

16. Created a generic config. file from which we can insert/change the values.

17. SonarQube execution with all project-level configurations in the separated config file.

18. Gulp composer implemented for downloading composer dependencies. Also automated installation of composer on runtime.

19. Integrated Gulp-bower.

20. Implemented Gulp CI on ALM\_TASK\_MANAGER dummy project.

21. Implemented PHPDox.

22.Provided training to team on Gulp,Gulp CI,Gulp-bower,browser-sync using GULP.

7)Nodejs

-Explored and performed development in nodejs

-Performed basic nodejs CI gulp Gulp as the build tool.

8)Functional programming

-Attended sessions of functional programming

-Watched videos and also explored and performed development in functional programming

9)AntCI-Bamboo

-Performed unit-testing,static-code analysis,code-coverage,packaging and also reports generation of each in bamboo on a php code-base using ANT as the build tool for automation.

-same as python-ci bamboo

10)SONAR

- Performed SonarQube implementation for "Dashboard Management" team

- provided KT about the UI and also the workflow of sonar.

11)Media-trafficking

Solved issues of Mediatrafficking team with the file exclusions in SonarQube they are using and also provided information in test-case execution in bamboo for a python project.

12)PT-Portal

-Worked on pull-request in Stash and successfully executed integrating Stash with Jenkins.

-Explored Github for an alternative to Stash.

-Worked on the proposal-demo created for the PT-Portal team.

13)SCALA(Update)

-Exploring and learning the scala language.

-Created demo project in scala.

-Executed basic CI using SBT as build-tool.

-Performed R&D on the tools scalatest,scoverage,codeacy,scalariform,scapegoat and implemented the same.

-Performed basic scala CI on the dummy project created.

14)Docker

-watching videos

-watched cd workflow tutum-AWS video.

-INSTALLED docker and upgraded it

-set up docker registry

-pocs done

1. Built an image and performed the following in it

-tomcat installation and set up users using run.sh

-jdk installation

-deployed app.war to tomcat server i.e at /opt/tomcat/webapps/ROOT.war

1. Port forwarding using EXPOSE and the –p option i.e

docker run –d –p 8081:8080 pallavi/test

1. Container linking

-one container has a java app(EMS\_Hibernate.war) deployed at /opt/tomcat/webapps/

-2nd container is has sql installed and db scripts of the app in it.

-Set the gateway IP through which docker listens the sql container alongwith desired port in the jdbc-properties.conf in the war of the application.(standard is to set the base machine i.e VM’s ip)

-browser url localhost:8081/EMS\_Hibernate/home.jsp

1. Dockerising a node js app
2. Setup Docker UI

-pocs TO DO

1. Run NodeJS anywhere using Docker and Nexe
2. Dockerising a PHP app
3. Bamboo-Docker integration
4. Dockerizing your Scala apps with sbt-docker
5. Running Jenkins in Docker Containers
6. Shipping nodejs apps with Docker and code-shipping.

15)working in detail on scalariform customization using git - hooks concept.

16)Media-trafficking team-kavitanjali

Solved issues in sonarqube as well as sonar-runner execution

Details in D:\D-Data\PALLAVI\ALM\ALM Tasks\SONAR\media-trafficking issue\

17)Mongodb

-training attended

-executed basic commands of CRUD

-Scala integrated with mongodb using casbah driver

18)Integration of Alm tools- Jenkins,Stash,Jira,Confluence

19)Exploration of Jira,Confluence

20)Bamboo exploration and implementation of concepts in details

Note:details at D:\D-Data\PALLAVI\ALM\ALM Tasks\PT-PORTAL\Ptportal-tools-integration

-branching and merging

-artifacts promotion

-permissions

-basic deployment

TO DO

-svn

-fisheye-crucible

21)Docker Compose basic POC done

Container linking - take a Java application and link two containers i.e. tomcat and mysql using compose.

22) Apache Spark

-Attended training

-Attended training by cloud coe team

-videos

<http://learn.mapr.com/dev-360-apache-spark-essentials>

<https://www.youtube.com/watch?v=yzfCTNukfl8>

<https://app.pluralsight.com/library/courses/apache-spark-fundamentals/table-of-contents>

-written blog in medium as DipshikhaTutun

<https://medium.com/@DipshikhaTutun/before-diving-into-apache-spark-do-keep-in-mind-the-following-d885a44c84f7#.bruhef4ru>